

Building Ice for C++ with MinGW

This page describes the Ice source distribution, including information about compiler requirements, third-party dependencies, and instructions for building and testing the distribution.



MinGW is only supported for building the Ice extension for Ruby. It is not supported for general application development. If you prefer, you can download a [Windows installer](#) that contains a complete Ice for Ruby run-time.

On this page:

- [C++ Build Requirements for MinGW](#)
 - [MinGW Compiler](#)
 - [Third-party Libraries](#)
- [Compiling and Testing Ice for C++ with MinGW](#)
 - [Building Ice](#)
 - [Running the Test Suite](#)
- [Installing a C++ Source Build for MinGW](#)

C++ Build Requirements for MinGW

Ice for C++ was extensively tested using the operating systems and compiler versions listed on our [platforms page](#).

MinGW Compiler

The only supported MinGW version is the one included in the [Ruby Development Kit 4.5.2](#). You must download the file `DevKit-tdm-32-4.5.2-20111229-1559-sfx.exe`, which is a self-extracting archive.

Third-party Libraries

Ice has dependencies on a number of third-party libraries:

- [OpenSSL 1.0](#) or later
- [bzip2 1.0](#)
- [mcpp 2.7.2](#) (with patches)

You do not need to build these packages yourself, as ZeroC supplies a [Windows installer](#) that contains release and debug libraries for all of the third-party dependencies.

If you intend to build the third-party dependencies from source, we recommend downloading the [Ice third-party source archive](#). This archive contains the source distributions for each of the third-party dependencies, as well as required source patches and configuration instructions.

Compiling and Testing Ice for C++ with MinGW

Building Ice

Using your favorite Zip tool, unzip the Ice source archive anywhere you like.

Open a Windows command prompt and change the working directory to the location of the extracted source archive. For example:

```
> cd C:\Ice-3.5.1\cpp
```

Run the `devkitvars.bat` batch file from the Ruby development kit to setup your environment. For example, if you installed the development kit in `C:\RubyDevKit-4.5.2`, run the following:

```
> C:\RubyDevKit-4.5.2\devkitvars.bat
```

Edit `config\Make.rules` to establish your build configuration. The comments in the file provide more information. In particular, if Ice third-party packages are not installed in the default location, set `THIRDPARTY_HOME` to the Ice third-party installation directory.

Now you are ready to build Ice:

```
> make
```

This will build the C++ and Ruby slice translators, the Ice core, IceSSL and C++ tests.

Running the Test Suite

`Python` is required to run the test suite. After a successful build, you can run the tests as follows:

```
> python allTests.py
```

If everything worked out, you should see lots of "ok" messages. In case of a failure, the tests abort with "failed".

Installing a C++ Source Build for MinGW

Simply run `make install` from a command prompt. This will install Ice in the directory specified by the `prefix` variable in `config\Make.rules`.